Generalized Feedback Vertex Set Problems on Bounded-Treewidth Graphs: Chordality Is the Key to Single-Exponential Parameterized Algorithms*†

Édouard Bonnet[1], Nick Brettell[2], O-joung Kwon[3], and Dániel Marx[4]

1 **Department of Computer Science, Middlesex University, London, UK**
2 **School of Mathematics and Statistics, Victoria University of Wellington, Wellington, New Zealand**
3 **Logic and Semantics, Technische Universität Berlin, Berlin, Germany**
4 **Institute for Computer Science and Control, Hungarian Academy of Sciences, (MTA SZTAKI), Budapest, Hungary**

Abstract

It has long been known that Feedback Vertex Set can be solved in time $2^{\mathcal{O}(w \log w)} n^{\mathcal{O}(1)}$ on graphs of treewidth w, but it was only recently that this running time was improved to $2^{\mathcal{O}(w)} n^{\mathcal{O}(1)}$, that is, to single-exponential parameterized by treewidth. We investigate which generalizations of Feedback Vertex Set can be solved in a similar running time. Formally, for a class of graphs $\mathcal{P}$, Bounded $\mathcal{P}$-Block Vertex Deletion asks, given a graph G on n vertices and positive integers k and d, whether G contains a set S of at most k vertices such that each block of $G - S$ has at most d vertices and is in $\mathcal{P}$. Assuming that $\mathcal{P}$ is recognizable in polynomial time and satisfies a certain natural hereditary condition, we give a sharp characterization of when single-exponential parameterized algorithms are possible for fixed values of d:

- if $\mathcal{P}$ consists only of chordal graphs, then the problem can be solved in time $2^{\mathcal{O}(wd^2)} n^{\mathcal{O}(1)}$,
- if $\mathcal{P}$ contains a graph with an induced cycle of length $\ell \geqslant 4$, then the problem is not solvable in time $2^{o(w \log w)} n^{\mathcal{O}(1)}$ even for fixed $d = \ell$, unless the ETH fails.

We also study a similar problem, called Bounded $\mathcal{P}$-Component Vertex Deletion, where the target graphs have connected components of small size instead of having blocks of small size, and present analogous results.

1998 ACM Subject Classification G.2.1 Combinatorial Algorithms, G.2.2 Graph Algorithms

Keywords and phrases fixed-parameter tractable algorithms, treewidth, feedback vertex set

Digital Object Identifier 10.4230/LIPIcs.IPEC.2017.7

1 Introduction

Treewidth is a measure of how well a graph accommodates a decomposition into a tree-like structure. In the field of parameterized complexity, many NP-hard problems have been shown to have FPT algorithms when parameterized by treewidth; for example, Coloring, Vertex Cover, Feedback Vertex Set, and Steiner Tree. In fact, Courcelle [6] established a

* All authors were supported by ERC Starting Grant PARAMTIGHT (No. 280152) and ERC Consolidator Grant SYSTEMATICGRAPH (No. 725978). The third author was also supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (ERC consolidator grant DISTRUCT, agreement No. 648527).
† The full version can be found in [5], `https://arxiv.org/abs/1704.06757`.

12th International Symposium on Parameterized and Exact Computation (IPEC 2017).
Editors: Daniel Lokshtanov and Naomi Nishimura; Article No. 7; pp. 7:1–7:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

meta-theorem that every problem definable in MSO_2 logic can be solved in linear time on graphs of bounded treewidth. While Courcelle's Theorem is a very general tool for obtaining algorithmic results, for specific problems dynamic programming techniques usually give algorithms where the running time $f(w)n^{\mathcal{O}(1)}$ has better dependence on treewidth w. There is some evidence that careful implementation of dynamic programming (plus maybe some additional ideas) gives optimal dependence for some problems (see, e.g., [12]).

For Feedback Vertex Set, standard dynamic programming techniques give $2^{\mathcal{O}(w \log w)} n^{\mathcal{O}(1)}$-time algorithms and it was considered plausible that this could be the best possible running time. Hence it was a remarkable surprise when it turned out that $2^{\mathcal{O}(w)}n^{\mathcal{O}(1)}$ algorithms are also possible for this problem by various techniques: Cygan et al. [7] obtained a $3^w n^{\mathcal{O}(1)}$-time randomized algorithm by using the so-called Cut & Count technique, and Bodlaender et al. [2] showed there is a deterministic $2^{\mathcal{O}(w)}n^{\mathcal{O}(1)}$-time algorithm by using a rank-based approach and the concept of representative sets. This was also later shown in the more general setting of representative sets in matroids by Fomin et al. [11].

Generalized feedback vertex set problems. We explore the extent to which these results apply for generalizations of Feedback Vertex Set. The Feedback Vertex Set problem asks for a set S of at most k vertices such that $G - S$ is acyclic, or in other words, every block of $G - S$ is a single edge or vertex. We consider generalizations where we allow the blocks to be some other type of small graph, such as triangles, small cycles, or small cliques; these generalizations were first studied in [4]. The main result of this paper is that the existence of single-exponential algorithms is closely linked to whether the small graphs we are allowing are all chordal or not. Formally, we consider the following problem:

Bounded $\mathcal{P}$-Block Vertex Deletion **Parameter:** d, w
Input: A graph G of treewidth at most w, and positive integers d and k.
Question: Is there a set S of at most k vertices in G such that each block of $G - S$ has at most d vertices and is in $\mathcal{P}$?

The result of Bodlaender et al. [2] implies that when $d = 2$, Bounded $\mathcal{P}$-Block Vertex Deletion can be solved in time $2^{\mathcal{O}(w)}n^{\mathcal{O}(1)}$. Our main question is for which graph classes $\mathcal{P}$ can this problem be solved in time $2^{\mathcal{O}(w)}n^{\mathcal{O}(1)}$, when we regard d as a fixed constant. A graph is *chordal* if it has no induced cycles of length at least 4. We show that if $\mathcal{P}$ consists of only chordal graphs, then we can solve this problem in single-exponential time for fixed d.

▶ **Theorem 1.** *Let $\mathcal{P}$ be a class of graphs that is block-hereditary, recognizable in polynomial time, and consists of only chordal graphs. Then* Bounded $\mathcal{P}$-Block Vertex Deletion *can be solved in time $2^{\mathcal{O}(wd^2)}k^2 n$ on graphs with n vertices and treewidth w.*

The condition that $\mathcal{P}$ is block-hereditary ensures that the class of graphs with blocks in $\mathcal{P}$ is hereditary; a formal definition is given in Section 2. We complement this result by showing that if $\mathcal{P}$ contains a graph that is not chordal, then single-exponential algorithms are not possible (assuming ETH), even for fixed d. Note that if $\mathcal{P}$ is block-hereditary and contains a graph that is not chordal, then this graph contains a chordless cycle on $\ell \geqslant 4$ vertices and consequently the cycle graph on ℓ vertices is also in $\mathcal{P}$.

▶ **Theorem 2.** *If $\mathcal{P}$ contains the cycle graph on $\ell \geqslant 4$ vertices, then* Bounded $\mathcal{P}$-Block Vertex Deletion *is not solvable in time $2^{o(w \log w)}n^{\mathcal{O}(1)}$ on graphs of treewidth at most w even for fixed $d = \ell$, unless the ETH fails.*

Baste et al. [1] recently studied the complexity of a similar problem, where the task is to find a set of vertices whose deletion results in a graph with no minor in a given collection of graphs $\mathcal{F}$, parameterized by treewidth. When $\mathcal{F} = \{C_4\}$, this is equivalent to Bounded $\mathcal{P}$-Block Vertex Deletion where $\mathcal{P} = \{K_2, K_3\}$, and the complexity they obtain in this case is consistent with our result.

Whether this lower bound of Theorem 2 is best possible when $\mathcal{P}$ contains a cycle on $\ell \geqslant 4$ vertices remains open. However, as partial evidence towards this, we note that when $\mathcal{P}$ contains all graphs, the result by Baste et al. [1] implies that that Bounded $\mathcal{P}$-Block Vertex Deletion can be solved in time $2^{\mathcal{O}(w \log w)} n^{\mathcal{O}(1)}$ when d is fixed, as the minor obstruction set $\mathcal{F}$ consists of all of 2-connected graphs with $d+1$ vertices.

Bounded-size components. Using a similar technique, we can obtain analogous results for a slightly simpler problem, that we call Bounded $\mathcal{P}$-Component Vertex Deletion, where we want to remove at most k vertices such that each connected component of the resulting graph has at most d vertices and belongs to $\mathcal{P}$. If we have only the size constraint (i.e., $\mathcal{P}$ contains every graph), then this problem is known as Component Order Connectivity [9]. Drange et al. [9] studied the parameterized complexity of a weighted variant of the Component Order Connectivity problem; their results imply, in particular, that Component Order Connectivity can be solved in time $2^{\mathcal{O}(k \log d)} n$, but is $W[1]$-hard parameterized by only k or d. The corresponding edge-deletion problem, parameterized by treewidth, was studied by Enright and Meeks [10].

▶ **Theorem 3.** *Let $\mathcal{P}$ be a class of graphs that is hereditary, recognizable in polynomial time, and consists of only chordal graphs. Then* Bounded $\mathcal{P}$-Component Vertex Deletion *can be solved in time $2^{\mathcal{O}(wd^2)} k^2 n$ on graphs with n vertices and treewidth w.*

▶ **Theorem 4.** *If $\mathcal{P}$ contains the cycle graph on $\ell \geqslant 4$ vertices, then* Bounded $\mathcal{P}$-Component Vertex Deletion *is not solvable in time $2^{o(w \log w)} n^{\mathcal{O}(1)}$ on graphs of treewidth at most w even for fixed $d = \ell$, unless the ETH fails.*

The result of Baste et al. [1] implies that when $\mathcal{P}$ contains all graphs, Bounded $\mathcal{P}$-Component Vertex Deletion can be solved in time $2^{\mathcal{O}(w \log w)} n^{\mathcal{O}(1)}$. When d is not fixed, one might ask whether Bounded $\mathcal{P}$-Component Vertex Deletion admits an $f(w)n^{\mathcal{O}(1)}$-time algorithm; that is, an FPT algorithm parameterized only by treewidth. We provide a negative answer: the problem is $W[1]$-hard when $\mathcal{P}$ contains all chordal graphs, even parameterized by both treewidth and k. Furthermore, two stronger lower bound results hold, under the assumption of the ETH.

▶ **Theorem 5.** *Let $\mathcal{P}$ be a hereditary class containing all chordal graphs. Then* Bounded $\mathcal{P}$-Component Vertex Deletion *is $W[1]$-hard parameterized by the combined parameter (w, k). Moreover, unless the ETH fails, (1) this problem has no $f(w)n^{o(w)}$-time algorithm; and (2) it has no $f(k')n^{o(k'/\log k')}$-time algorithm, where $k' = w + k$.*

Techniques. A pair (G, S) consisting of a graph G and a vertex subset S of G will be called a boundaried graph, and an S-block of G is a block of G containing an edge with both endpoints in S. The algorithm for Bounded $\mathcal{P}$-Block Vertex Deletion uses several lemmas on S-blocks of boundaried graphs (G, S), which appear in Section 3. The key property is the following: (*) when we merge two boundaried graphs (G, S) and (H, S) into a graph G', to decide whether each S-block of G' is some fixed target graph that is chordal, it is sufficient to know, for each non-trivial block B of $G[S]$ or $H[S]$, some local information about B in the S-block containing B in G or H, respectively. We think of target graphs as labeled graphs where any two vertices in the same block have distinct labels in

$\{1, \ldots, d\}$, and the local information referred to in (*) is the set of labels of neighbors of B in the S-block containing B. The related result is stated as Proposition 6. This will be used to determine whether each of the S-blocks of G' is one of the target graphs in $\mathcal{P}$. After then, to decide whether G' is a required graph, it remains to check that the whole graph has no chordless cycle, since there is a possibility of linking two controlled blocks by a sequence of uncontrolled blocks in both sides G and H, and thus creating a chordless cycle in G'. This second part can be dealt with in a similar manner to the single-exponential time algorithm for Feedback Vertex Set, using representative-set techniques.

2 Preliminaries

We follow the terminology of Diestel [8], unless otherwise specified. A vertex v of G is a *cut vertex* if the deletion of v from G increases the number of connected components. We say G is *biconnected* if it is connected and has no cut vertices. Note that every connected graph on at most two vertices is biconnected. A *block* of G is a maximal biconnected subgraph of G. We say G is 2-*connected* if it is biconnected and $|V(G)| \geqslant 3$. An induced cycle of length at least four is called a *chordless cycle*. A graph is *chordal* if it has no chordless cycles. For a class of graphs $\mathcal{P}$, a graph is called a $\mathcal{P}$-*block graph* if each of its blocks is in $\mathcal{P}$. A class $\mathcal{C}$ of graphs is *block-hereditary* if for every $G \in \mathcal{C}$ and every biconnected induced subgraph H of G, $H \in \mathcal{C}$. For two integers d_1, d_2 with $d_1 \leqslant d_2$, let $[d_1, d_2]$ be the set of all integers i with $d_1 \leqslant i \leqslant d_2$, and for a positive integer, let $[d] := [1, d]$. For a function $f : X \to Y$ and $X' \subseteq X$, the function $f' : X' \to Y$ where $f'(x) = f(x)$ for all $x \in X'$ is called the *restriction* of f on X', and is denoted $f|_{X'}$. We also say that f *extends* f' to the set X.

Block d-labeling. A *block d-labeling* of a graph G is a function $L : V(G) \to [d]$ such that for each block B of G, $L|_{V(B)}$ is an injection. If G is equipped with a block d-labeling L, then it is called a *(block) d-labeled graph*, and we call $L(v)$ the *label* of v. Two d-labeled graphs G and H are *label-isomorphic* if there is a graph isomorphism from G to H that is label preserving. For biconnected block d-labeled graphs G and H, H is *partially label-isomorphic* to G if H is label-isomorphic to the subgraph of G induced by the vertices with labels in H.

Treewidth. A *tree decomposition* of a graph G is a pair $(T, \mathcal{B})$ consisting of a tree T and a family $\mathcal{B} = \{B_t\}_{t \in V(T)}$ of sets $B_t \subseteq V(G)$, called *bags*, satisfying the following three conditions: (1) $V(G) = \bigcup_{t \in V(T)} B_t$, (2) for every edge uv of G, there exists a node t of T such that $u, v \in B_t$, (3) for $t_1, t_2, t_3 \in V(T)$, $B_{t_1} \cap B_{t_3} \subseteq B_{t_2}$ whenever t_2 is on the path from t_1 to t_3 in T. The *width* of a tree decomposition $(T, \mathcal{B})$ is $\max\{|B_t| - 1 : t \in V(T)\}$. The *treewidth* of G is the minimum width over all tree decompositions of G. A tree decomposition $(T, \mathcal{B} = \{B_t\}_{t \in V(T)})$ is *nice* if T is a rooted tree with root node r, and every node t of T is one of the following: (1) a *leaf node*: t is a leaf of T and $B_t = \emptyset$; (2) an *introduce node*: t has exactly one child t' and $B_t = B_{t'} \cup \{v\}$ for some $v \in V(G) \setminus B_{t'}$; (3) a *forget node*: t has exactly one child t' and $B_t = B_{t'} \setminus \{v\}$ for some $v \in B_{t'}$; or (4) a *join node*: t has exactly two children t_1 and t_2, and $B_t = B_{t_1} = B_{t_2}$.

Boundaried graphs. For a graph G and $S \subseteq V(G)$, the pair (G, S) is a *boundaried graph*. When G is a d-labeled graph, we simply say that (G, S) is a d-labeled graph. Two d-labeled graphs (G, S) and (H, S) are said to be *compatible* if $V(G - S) \cap V(H - S) = \emptyset$, $G[S] = H[S]$, and G and H have the same labels on S. For two compatible d-labeled graphs (G, S) and (H, S), the *sum* of two graphs $(G, S) \oplus (H, S)$ is the graph obtained from the disjoint union of

G and H by identifying each vertex in S and removing an edge if multiple edges appear. We denote by $L_G \oplus L_H$ the function from $V((G,S) \oplus (H,S))$ to $[d]$ where for $v \in V(G) \cup V(H)$, $(L_G \oplus L_H)(v) = L_G(v)$ if $v \in V(G)$ and $(L_G \oplus L_H)(v) = L_H(v)$ otherwise. For two unlabeled boundaried graphs, we define the sum in the same way, but ignoring the label condition.

A block of a graph is *non-trivial* if it has at least two vertices. For a boundaried graph (G,S), a block B of G is called an *S-block* if it contains an edge of $G[S]$. Note that every non-trivial block of $G[S]$ is contained in a unique S-block of G because two distinct blocks share at most one vertex. Let (G,S) be a boundaried graph. We define $\mathbf{Aux}(G,S)$ as the bipartite boundaried graph with bipartition $(\mathcal{C}_1, \mathcal{C}_2)$ and boundary $\mathcal{C}_2$ such that (1) $\mathcal{C}_1$ is the set of components of G, and $\mathcal{C}_2$ is the set of components of $G[S]$, (2) for $C_1 \in \mathcal{C}_1$ and $C_2 \in \mathcal{C}_2$, $C_1C_2 \in E(\mathbf{Aux}(G,S))$ if and only if C_2 is contained in C_1. When (G,S) and (H,S) are two compatible d-labeled graphs, $\mathbf{Aux}(G,S) \oplus \mathbf{Aux}(H,S)$ is well-defined, as G and H have the same set of components on S. For a set S and a set $\mathcal{X}$ of subsets of S, let $\mathbf{Inc}(S,\mathcal{X})$ be the bipartite graph on the bipartition $(S,\mathcal{X})$ where for $v \in S$ and $X \in \mathcal{X}$, v and X are adjacent in $\mathbf{Inc}(S,\mathcal{X})$ if and only if $v \in X$. For a boundaried graph (G,S), when $\mathcal{P}$ is the partition of the set $\mathcal{C}$ of components of $G[S]$ such that two components of $G[S]$ are in the same part if and only if they are in the same component of G, we denote by $\mathbf{Inc}(\mathcal{C},\mathcal{P}) \sim \mathbf{Aux}(G,S)$.

3 Lemmas about S-blocks

We present several lemmas regarding S-blocks. For a biconnected d-labeled graph Q, a d-labeled graph (G,S) is *block-wise partially label-isomorphic to Q* if every S-block B of G is partially label-isomorphic to Q. For two compatible d-labeled graphs (G,S) and (H,S) with labelings L_G and L_H respectively, we say (G,S) and (H,S) are *block-wise Q-compatible* if

1. (G,S) and (H,S) are block-wise partially label-isomorphic to Q; and
2. for every non-trivial block B of $G[S]$, letting B_1 and B_2 be the S-blocks of G and H that contain B, respectively, $L_G(N_{B_1}(V(B)) \setminus S) \cap L_H(N_{B_2}(V(B)) \setminus S) = \emptyset$, and, for $\ell_1 \in L_G(N_{B_1}(V(B)) \setminus S)$ and $\ell_2 \in L_H(N_{B_2}(V(B)) \setminus S)$, the vertices in Q with labels ℓ_1 and ℓ_2 are not adjacent.

We describe sufficient conditions for when, given a chordal labeled graph Q, the sum of two given labeled graphs (G,S) and (H,S), each partially label-isomorphic to Q, is also partially label-isomorphic to Q.

▶ **Proposition 6.** *Let Q be a biconnected d-labeled chordal graph. Let (G,S) and (H,S) be two block-wise Q-compatible d-labeled graphs such that $\mathbf{Aux}(G,S) \oplus \mathbf{Aux}(H,S)$ has no cycles. Then $(G,S) \oplus (H,S)$ is block-wise partially label-isomorphic to Q.*

We use the following essential property of chordal graphs.

▶ **Lemma 7.** *Let F be a connected graph and let Q be a connected chordal graph. Let $\mu : V(F) \to V(Q)$ be a function such that for every induced path $p_1 \cdots p_m$ in F of length at most two, $\mu(p_1), \dots, \mu(p_m)$ are pairwise distinct and $\mu(p_1) \cdots \mu(p_m)$ is an induced path of Q. Then μ is an injection and preserves the adjacency relation.*

▶ **Lemma 8.** *Let (G,S) and (H,S) be two compatible d-labeled graphs such that $\mathbf{Aux}(G,S) \oplus \mathbf{Aux}(H,S)$ has no cycles. (1) If F is an S-block of $(G,S) \oplus (H,S)$ and uv is an edge in F, then uv is contained in some S-block of G or H. (2) Suppose each S-block of G or H is chordal. If F is an S-block of $(G,S) \oplus (H,S)$ and uvw is an induced path in F such that u and w are not contained in the same S-block of G or H, then $v \in S$, and there is an induced path $q_1q_2 \cdots q_\ell$ from $u = q_1$ to $w = q_\ell$ in $F - v$ such that each q_i is a neighbor of v.*

Proof of Proposition 6. Let F be an S-block of $(G,S)\oplus(H,S)$. Let L_G and L_H be labelings of G and H, respectively, and let $L := L_G \oplus L_H$. We may assume $|V(F)| \geqslant 3$. By Lemma 8, every edge of F is contained in some S-block of G or H. Thus, for $uv \in E(F)$, we have $L(u) \neq L(v)$ and the vertices with labels $L(u)$ and $L(v)$ are adjacent in Q. Moreover, since (G,S) and (H,S) are block-wise partially label-isomorphic to Q, we have $L(V(F)) \subseteq L_Q(V(Q))$. Let $\mu : V(F) \to V(Q)$ such that for each $v \in V(F)$, $L(v) = L_Q(\mu(v))$.

To apply Lemma 7, it is sufficient to prove that if uvw is an induced path in F, then $L(u) \neq L(w)$ and $\mu(u)\mu(v)\mu(w)$ is an induced path in Q. Since (G,S) and (H,S) are block-wise partially label-isomorphic to Q, if all of u,v,w are contained in an S-block of G or H, then it follows from the given condition. We may assume u and w are not contained in the same S-block of G or H. Then by (2) of Lemma 8, $v \in S$, and there is an induced path $q_1q_2\cdots q_\ell$ from $u = q_1$ to $w = q_\ell$ in $F - v$ such that each q_i is a neighbor of v.

We show that for $i \in \{1,\ldots,\ell-2\}$, $L(q_i), L(q_{i+1}), L(q_{i+2})$ are pairwise distinct, and $\mu(q_i)\mu(q_{i+1})\mu(q_{i+2})$ is an induced path of Q. If all of q_i, q_{i+1}, q_{i+2} are contained in G or H, then they are contained in the same S-block as v, and the claim follows. We may assume q_i and q_{i+2} are in distinct graphs of $G - S$ and $H - S$. Then the S-block containing q_i, q_{i+1}, v and the S-block containing q_{i+1}, q_{i+2}, v share the edge $q_{i+1}v$. Since (G,S) and (H,S) are block-wise Q-compatible, $L(q_i) \neq L(q_{i+2})$ and $\mu(q_i)$ is not adjacent to $\mu(q_{i+2})$ in Q.

We verify that $\mu(q_1)\mu(q_2)\cdots\mu(q_\ell)$ is an induced path of Q. Suppose this is false, and choose $i_1, i_2 \in \{1,2,\ldots,\ell\}$ with $i_2 - i_1 > 1$ and minimum $i_2 - i_1$ such that $\mu(q_{i_1})$ is adjacent to $\mu(q_{i_2})$ in Q. By minimality, $\mu(q_{i_1})\cdots\mu(q_{i_2-1})$ and $\mu(q_{i_1+1})\cdots\mu(q_{i_2})$ are induced paths and have length at least 2. Thus $\mu(q_{i_1})\cdots\mu(q_{i_2})$ is an induced cycle of length at least 4, contradicting the assumption that Q is chordal. Therefore, $\mu(q_1)\mu(q_2)\cdots\mu(q_\ell)$ is an induced path of Q, and, in particular, $L(u) \neq L(w)$ and $\mu(u)$ and $\mu(w)$ are not adjacent in Q, as required. By Lemma 7, we conclude that F is partially label-isomorphic to Q. ◀

Using Lemma 8, we can also prove the following.

▶ **Lemma 9.** *Let A be a set, let (G,S) and (H,S) be two compatible d-labeled graphs, and let $\mathcal{B}$ be the set of non-trivial blocks in $G[S]$. Suppose $g : \mathcal{B} \to A$ is a function where each S-block of G or H is chordal, $\mathbf{Aux}(G,S) \oplus \mathbf{Aux}(H,S)$ has no cycles, and for every $B_1, B_2 \in \mathcal{B}$ where B_1 and B_2 are contained in an S-block of G or H, $g(B_1) = g(B_2)$. If F is an S-block of $(G,S) \oplus (H,S)$ and $B_1, B_2 \in \mathcal{B}$ where $V(B_1), V(B_2) \subseteq V(F)$, then $g(B_1) = g(B_2)$.*

▶ **Proposition 10.** *Let (G,S) and (H,S) be two compatible d-labeled graphs such that every S-block of $(G,S) \oplus (H,S)$ is chordal. Then $(G,S) \oplus (H,S)$ is chordal if and only if $\mathbf{Aux}(G,S) \oplus \mathbf{Aux}(H,S)$ has no cycles.*

Proof. We briefly sketch the proof of one direction. Suppose that $\mathbf{Aux}(G,S) \oplus \mathbf{Aux}(H,S)$ has a cycle $C_1 - A_1 - C_2 - A_2 - \cdots - C_n - A_n - C_1$ where $C_1,\ldots,C_n$ are components of $G[S]$. For each $i \in \{1,\ldots,n\}$, let P_i be the shortest path from C_i to C_{i+1} in A_i, and let v_i, w_i be the end vertices of P_i where $v_i \in V(C_i)$ and $w_i \in V(C_{i+1})$. Let Q_i be the shortest path from w_i to v_{i+1} in C_{i+1}. We may assume $n \geqslant 3$; it is easy when $n = 2$. Then $v_1P_1 - Q_1 - P_2 - Q_2 - \cdots - P_n - Q_nv_1$ is a cycle in $(G,S) \oplus (H,S)$, but is not necessarily a chordless cycle. We claim that it contains a chordless cycle. Let x be the vertex following v_2 in P_2, and let y be the vertex preceding w_n in P_n. Take a shortest path P from x to y in the path $y - Q_n - P_1 - Q_1 - x$. Clearly P has length at least 2, as x and y are contained in distinct connected components of G or H. Also, every internal vertex of P has no neighbors in the other path of the cycle $v_1P_1 - Q_1 - P_2 - Q_2 - \cdots - P_n - Q_nv_1$ between x and y. So, if we take a shortest path P' from x to y along the other part of the cycle $v_1P_1 - Q_1 - P_2 - Q_2 - \cdots - P_n - Q_nv_1$, then $P \cup P'$ is a chordless cycle. ◀

4 Bounded $\mathcal{P}$-Block Vertex Deletion

We prove Theorem 1. We first focus on S-blocks of boundaried graphs (G, S). For each non-trivial block of $G[S]$, we guess its final shape, as a d-labeled biconnected graph, and store the labelings of the vertices and their neighbors in the S-block of G containing it. Collectively, we call this information a *characteristic* of (G, S). Using characteristics, we control S-blocks in $(G, S) \oplus (H, S)$, where (H, S) is a compatible d-labeled graph. By the previous step, we may assume that every S-block of $(G, S) \oplus (H, S)$ is in $\mathcal{P}$ and has at most d vertices. Note that $(G, S) \oplus (H, S)$ still may have a chordless cycle. By Proposition 10, if we assume that every S-block of $(G, S) \oplus (H, S)$ is in $\mathcal{P}$, then $(G, S) \oplus (H, S)$ is chordal if and only if $\mathbf{Aux}(G, S) \oplus \mathbf{Aux}(H, S)$ has no cycles. So, instead of keeping $\mathbf{Aux}(G, S)$, we store the corresponding partition of the set of components of $G[S]$.

For convenience, we fix an integer $d \geqslant 2$ and a class $\mathcal{P}$ of graphs that is block-hereditary, recognizable in polynomial time, and consists of only chordal graphs. Let $\mathcal{U}_d$ be the set of all d-labeled biconnected $\mathcal{P}$-block graphs, where each H in $\mathcal{U}_d$ has labeling L_H. For a boundaried graph (G, S), we denote by $\text{Block}(G, S)$ the set of all non-trivial blocks in $G[S]$.

For a d-labeled graph (G, S) with a labeling L, a *characteristic* of (G, S) is a pair (g, h) of functions $g : \text{Block}(G, S) \to \mathcal{U}_d$ and $h : \text{Block}(G, S) \to 2^{[d]}$ satisfying the following, for each $B \in \text{Block}(G, S)$ and the unique S-block H of G containing B,

1. (label-isomorphic condition) H is partially label-isomorphic to $g(B)$;
2. (coincidence condition) for every $B' \in \text{Block}(G, S)$ with $V(B') \subseteq V(H)$, $g(B') = g(B)$;
3. (neighborhood condition) $h(B) = L(N_H(V(B)) \setminus S)$; and
4. (complete condition) for every w where $w \in V(H) \setminus S$ or $\{w\} = V(H) \cap V(C)$ for some component C of $G[S]$, $H[N_H[w]]$ is label-isomorphic to $g(B)[N_{g(B)}[z]]$ where z is the vertex in $g(B)$ with label $L(w)$.

We say that the sum $(G, S) \oplus (H, S)$ *respects* (g, h) if for each $B \in \text{Block}(G, S)$, the S-block of $(G, S) \oplus (H, S)$ containing B is label-isomorphic to $g(B)$. The following is the main combinatorial result regarding characteristics.

▶ **Theorem 11.** *Let (G_1, S), (G_2, S), (H, S) be d-labeled $\mathcal{P}$-block graphs such that each (G_i, S) is compatible with (H, S), (G_1, S) and (G_2, S) have the same characteristic (g, h), and $\mathbf{Aux}(G_2, S) \oplus \mathbf{Aux}(H, S)$ has no cycles. If $(G_1, S) \oplus (H, S)$ is a d-labeled $\mathcal{P}$-block graph that respects (g, h), then $(G_2, S) \oplus (H, S)$ is a d-labeled $\mathcal{P}$-block graph that respects (g, h).*

Proof. We show $(G_2, S) \oplus (H, S)$ respects (g, h). Choose a non-trivial block B of $G_2[S]$, let $Q := g(B)$, let F be the S-block of $(G_2, S) \oplus (H, S)$ containing B, L_F be the function from $V(F)$ to $[d]$ that sends each vertex to its label from G_2 or H, and L_Q be the labeling of Q.

We claim that F is label-isomorphic to Q. We regard F as the sum of $(F \cap G_2, V(F) \cap S)$ and $(F \cap H, V(F) \cap S)$ and verify the conditions of Proposition 6. Using Lemma 9, for every $B' \in \text{Block}(G_2, S)$ with $V(B') \subseteq V(F)$, $g(B') = Q$. We also observe that $\mathbf{Aux}(F \cap G_2, S_F) \oplus \mathbf{Aux}(F \cap H, S_F)$ has no cycles as $\mathbf{Aux}(G_2, S) \oplus \mathbf{Aux}(H, S)$ has no cycles. Since (g, h) is a characteristic of (G_2, S) and $(G_1, S) \oplus (H, S)$ respects (g, h), we can confirm that both $F \cap G$ and $F \cap H$ are block-wise partially label-isomorphic to Q. The second condition of being block-wise Q-compatible follows from the fact that (G_1, S) and (G_2, S) have the same characteristic (g, h). Thus, $F \cap G_2$ and $F \cap H$ are block-wise Q-compatible, and this implies that F is partially label-isomorphic to Q by Proposition 6. By the 'complete condition' of a characteristic, we can show that $L_Q(V(Q)) \subseteq L_F(V(F))$, so F is label-isomorphic to Q.

Lastly, we can confirm that $(G_2, S) \oplus (H, S)$ is a d-labeled $\mathcal{P}$-block graph by showing that every non S-block of $(G_2, S) \oplus (H, S)$ is fully contained in G_2 or H. We can argue this using the fact that $(G_2, S) \oplus (H, S)$ is chordal, which is implied by Proposition 10. ◀

Proof of Theorem 1. We obtain a nice tree decomposition $(T, \mathcal{B} = \{B_t\}_{t \in V(T)})$ of G with root node r and width at most $5w+4$ in time $\mathcal{O}(c^w \cdot n)$ for some constant c using the approximation algorithm by Bodlaender et al. [3]. For $t \in V(T)$, let G_t be the subgraph of G induced by the union of all bags $B_{t'}$ where t' is a descendant of t. Let $\mathrm{Comp}(t, X)$ be the set of all components of $G[B_t \setminus X]$, and $\mathrm{Part}(t, X)$ be the set of all partitions of $\mathrm{Comp}(t, X)$.

For each node t of T, $X \subseteq B_t$, and a function $L : B_t \setminus X \to [d]$, we define $\mathcal{F}(t, X, L)$ as the set of all pairs (g, h) consisting of functions $g : \mathrm{Block}(t, X) \to \mathcal{U}_d$ and $h : \mathrm{Block}(t, X) \to 2^{[d]}$. We say that (g, h) is *valid*, if (1) L is a d-labeling of $G[B_t \setminus X]$, (2) for each $B \in \mathrm{Block}(t, X)$, B is partially label-isomorphic to $g(B)$, and (3) for each $B \in \mathrm{Block}(t, X)$, $L(V(B)) \cap h(B) = \emptyset$. For $i \in \{0, 1, \ldots, k\}$ and $(g, h) \in \mathcal{F}(t, X, L)$, let $c[t, (X, L, i, (g, h))]$ be the family of all partitions $\mathcal{X} \in \mathrm{Part}(t, X)$ satisfying the following property: there exist $S \subseteq V(G_t) \setminus B_t$ with $|S| = i$ and a d-labeling L' of $G_t - (X \cup S)$ where (1) $L = L'|_{B_t \setminus X}$, (2) $G_t - (X \cup S)$ is a $\mathcal{P}$-block graph, (3) (g, h) is a characteristic of $(G_t - (X \cup S), B_t \setminus X)$, and (4) $\mathbf{Inc}(\mathrm{Comp}(t, X), \mathcal{X}) \sim \mathbf{Aux}(G_t - (X \cup S), B_t \setminus X)$. Such a pair (S, L') is a *partial solution* with respect to $\mathcal{X}$.

The main idea is that instead of fully computing $c[t, M]$ for $M = (X, L, i, (g, h))$, we recursively enumerate a set $r[t, M]$ that may represent partial solutions for $c[t, M]$. Formally, for a subset $r[t, M] \subseteq c[t, M]$, we denote $r[t, M] \equiv c[t, M]$ if for every $\mathcal{X} \in c[t, M]$ and a partial solution (S, L') with respect to $\mathcal{X}$ and $S_{out} \subseteq V(G) \setminus V(G_t)$ where $G - (S \cup X \cup S_{out})$ is a d-labeled $\mathcal{P}$-block graph respecting (g, h), there exists $\mathcal{X}_1 \in r[t, M]$ and a partial solution (S', L'') with respect to $\mathcal{X}_1$ such that $G - (S' \cup X \cup S_{out})$ is a d-labeled $\mathcal{P}$-block graph respecting (g, h). By the definition of $r[t, M]$, the problem is a Yes-instance if and only if there exists $(X, L, i, (g, h))$ for the root node r with $|X| + i \leqslant k$ such that $r[r, (X, L, i, (g, h)] \neq \emptyset$.

Whenever we update $r[t, M]$, we confirm that $|r[t, M]| \leqslant w \cdot 2^{w-1}$. This will be the application of the representative set technique developed by Bodlaender et al. [2]. For a set S and a set $\mathcal{A}$ of partitions of S, a subset $\mathcal{A}'$ of $\mathcal{A}$ is called a *representative set* if for every $\mathcal{X}_1 \in \mathcal{A}$ and every partition $\mathcal{Y}$ of S where $\mathbf{Inc}(S, \mathcal{X}_1 \cup \mathcal{Y})$ has no cycles, there exists a partition $\mathcal{X}_2 \in \mathcal{A}'$ such that $\mathbf{Inc}(S, \mathcal{X}_2 \cup \mathcal{Y})$ has no cycles.

▶ **Proposition 12.** *Given a family $\mathcal{A}$ of partitions of a set S, one can output a representative set of $\mathcal{A}$ of size at most $|S| \cdot 2^{|S|-1}$ in time $\mathcal{A}^{\mathcal{O}(1)} 2^{\mathcal{O}(|S|)}$.*

We sketch how to update families $r[t, M]$ when t is an introduce node with child node t'. We may assume (g, h) is valid, otherwise $c[t, M] = \emptyset$.

Let v be the vertex in $B_t \setminus B_{t'}$. If $v \in X$, then $G_t - X = G_{t'} - (X \setminus \{v\})$ and $B_t \setminus X = B_{t'} \setminus (X \setminus \{v\})$. Thus, we can set $r[t, M] := r[t', (X \setminus \{v\}, L, i, (g, h))]$. We assume $v \notin X$, and let $L_{res} := L|_{B_{t'} \setminus X}$. For $(g, h) \in \mathcal{F}(t, X, L)$, a pair $(g', h') \in \mathcal{F}(t', X, L_{res})$ is called the *restriction* of (g, h) if (1) for $B_1 \in \mathrm{Block}(t', X)$ and $B_2 \in \mathrm{Block}(t, X)$ with $V(B_1) \subseteq V(B_2)$, $g'(B_1) = g(B_2)$, and if $v \in V(B_2)$, then every vertex in $g'(B_1)$ with label in $h'(B_1)$ is not adjacent to the vertex in $g'(B_1)$ with label $L(v)$, (2) for $B_1 \in \mathrm{Block}(t', X)$ and $B_2 \in \mathrm{Block}(t, X)$ with $V(B_1) \subseteq V(B_2)$ and $v \notin V(B_2)$, $h'(B_1) = h(B_2)$, and (3) for $B_2 \in \mathrm{Block}(t, X)$ containing v, $h(B_2) = \bigcup_{B_1 \in \mathrm{Block}(t', X), V(B_1) \subseteq V(B_2)} h(B_1)$.

▶ **Claim 13.** *For $\mathcal{X} \in \mathrm{Part}(t, X)$, $\mathcal{X} \in c[t, M]$ if and only if there exist a restriction (g', h') of (g, h) and $\mathcal{Y} \in c[t', (X, L_{res}, i, (g', h'))]$ such that (1) v has neighbors on at most one component in each part of $\mathcal{Y}$, and (2) if v has at least one neighbor in $G[B_t \setminus X]$, then $\mathcal{X}$ is the partition obtained from $\mathcal{Y}$ by, for parts $Y_1, \ldots, Y_m$ of $\mathcal{Y}$ containing components having a neighbor of v, removing all of $Y_1, \ldots, Y_m$ and adding a part that consists of all components of $G[B_t \setminus X]$ not contained in parts of $\mathcal{Y} \setminus \{Y_1, \ldots, Y_m\}$; and otherwise, $\mathcal{X} = \mathcal{Y} \cup \{\{v\}\}$.*

We update $r[t, M]$ as follows. Set $\mathcal{K} := \emptyset$. For a pair of functions (g', h'), we test whether (g', h') is a restriction of (g, h). Assume (g', h') is a restriction of (g, h). For each $\mathcal{Y} \in r[t', (X, L_{res}, i, (g', h'))]$, we check the two conditions for (g', h') and $\mathcal{Y}$ in Claim 13, and if they are satisfied, then add the set $\mathcal{X}$ described in Claim 13 to $\mathcal{K}$; otherwise, skip it. The whole procedure can be done in time $2^{\mathcal{O}(wd^2)}$. After we do this for all possible candidates, we take a representative set of $\mathcal{K}$ using Proposition 12, and assign the resulting set to $r[t, M]$.

We claim that $r[t, M] \equiv c[t, M]$. Let $G_{out} := G - (V(G_t) \setminus B_t)$, $\mathcal{X} \in c[t, M]$, and (S, L') be a partial solution with respect to $\mathcal{X}$, and suppose there exists $S_{out} \subseteq V(G) \setminus V(G_t)$ where $(G_t - (X \cup S), B_t \setminus X) \oplus (G_{out} - (X \cup S_{out}), B_t \setminus X)$ is a d-labeled $\mathcal{P}$-block graph respecting (g, h). Every $(B_{t'} \setminus X)$-block of $G - (S \cup X \cup S_{out})$ is chordal as such a block is a $(B_t \setminus X)$-block of $G - (S \cup X \cup S_{out})$. Since $G - (S \cup X \cup S_{out})$ is chordal, by Proposition 10, $\mathbf{Aux}(G_{t'} - (X \cup S), B_{t'} \setminus X) \oplus \mathbf{Aux}(G_{out} - (X \cup S_{out}), B_{t'} \setminus X)$ has no cycles. Let $M_{res} := (X, L_{res}, i, (g', h'))$. As $r[t', M_{res}] \equiv c[t', M_{res}]$, there exist $\mathcal{Y} \in r[t', M_{res}]$ and a partial solution (S', L'') with respect to $\mathcal{Y}$ such that $\mathbf{Inc}(\mathrm{Comp}(t', X), \mathcal{Y}) \sim \mathbf{Aux}(G_{t'} - (X \cup S'), B_{t'} \setminus X)$ has no cycles. By Theorem 11, $G - (S' \cup X \cup S_{out})$ is a d-labeled $\mathcal{P}$-block graph respecting (g, h).

By the procedure, $\mathcal{X}_1$ where $\mathbf{Inc}(\mathrm{Comp}(t, X), \mathcal{X}_1) \sim \mathbf{Aux}(G_t - (X \cup S'), B_t \setminus X)$ is added to $\mathcal{K}$. And there exist $\mathcal{X}_2 \in r[t, M]$ and a partial solution (S'', L''') with respect to $\mathcal{X}_2$ such that $G - (S'' \cup X \cup S_{out})$ is a d-labeled $\mathcal{P}$-block graph. Thus, $r[t, M] \equiv c[t, M]$.

Total running time. We denote $|V(G)|$ by n. Note that the number of nodes in T is $\mathcal{O}(wn)$. For fixed $t \in V(T)$, there are at most 2^{w+1} possible choices for $X \subseteq B_t$, and for fixed $X \subseteq B_t$, there are at most d^{w+1} possible functions L. Furthermore, the size of $\mathcal{F}(t, X, L)$ is bounded by $2^{\mathcal{O}(wd^2)}$. Thus, there are $\mathcal{O}(n \cdot k \cdot \max(2, d)^{w+1} \cdot 2^{\mathcal{O}(wd^2)})$ tables. In summary, the algorithm runs in time $\mathcal{O}(n \cdot k \cdot \max(2, d)^{w+1}) \cdot 2^{\mathcal{O}(wd^2)} \cdot k = 2^{\mathcal{O}(wd^2)} k^2 n$. ◀

5 Lower bound for fixed d

We showed that Bounded $\mathcal{P}$-Component Vertex Deletion and Bounded $\mathcal{P}$-Block Vertex Deletion admit single-exponential time algorithms parameterized by treewidth, whenever $\mathcal{P}$ is a class of chordal graphs. We now establish that, assuming the ETH, this is no longer the case when $\mathcal{P}$ contains a graph that is not chordal.

In the $k \times k$ Independent Set problem, one is given a graph $G = ([k] \times [k], E)$ over the k^2 vertices of a *k-by-k grid*. We denote by $\langle i, j \rangle$ with $i, j \in [k]$ the vertex of G in the i-th *row* and j-th *column*. The goal is to find an independent set of size k in G that contains exactly one vertex in each row. The Permutation $k \times k$ Independent Set problem is similar but with the additional constraint that the independent set should also contain exactly one vertex per column.

▶ **Theorem 14.** *If $\mathcal{P}$ contains the cycle graph on $\ell \geqslant 4$ vertices, then* Bounded $\mathcal{P}$-Component Vertex Deletion, *or* Bounded $\mathcal{P}$-Block Vertex Deletion, *is not solvable in time $2^{o(w \log w)} n^{\mathcal{O}(1)}$ on graphs of treewidth at most w even for fixed $d = \ell$, unless the ETH fails.*

Proof. To prove this theorem, we reduce from Permutation $k \times k$ Independent Set which, like Permutation $k \times k$ Clique, cannot be solved in time $2^{o(k \log k)} k^{\mathcal{O}(1)}$ unless the ETH fails [13]. Let $G = ([k] \times [k], E)$ be an instance of Permutation $k \times k$ Independent Set. We assume that $\forall h, i, j \in [k]$ with $h \neq i$, $\langle i, j \rangle \langle h, j \rangle \in E$. Adding these edges does not change the Yes- and No-instances, but has the virtue of making Permutation $k \times k$ Independent Set equivalent to $k \times k$ Independent Set. We also assume that $\forall h, i, j \in [k]$, $\langle i, j \rangle \langle i, h \rangle \notin E$,

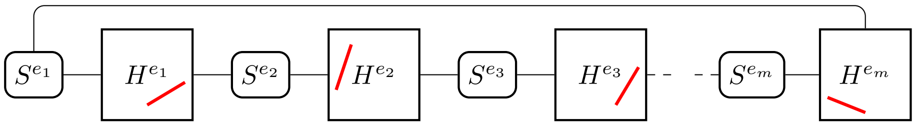

Figure 1 A high-level schematic of G' and G''. The H^{e_i}s only differ by a constant number of edges (in red/light gray) that encode their edge e_i of G.

since at most one of $\langle i, j\rangle$ and $\langle i, h\rangle$ can be in a given solution. Let $m := |E| = \mathcal{O}(k^4)$ be the number of edges of G.

Outline. We build two graphs $G' = (V', E')$ and $G'' = (V', E'')$ with treewidth at most $(3d+4)k+6d-5 = \mathcal{O}(k)$, and $((3d-2)k^2+2k)m$ vertices, where the following are equivalent:

1. G has an independent set of size k with one vertex per row of G.
2. There is a set $S \subseteq V'$ of size at most $(3d-2)k(k-1)m$ such that each connected component of $G' - S$ has size at most d and belongs to $\mathcal{P}$.
3. There is a set $S \subseteq V'$ of size at most $(3d-2)k(k-1)m$ such that each block of $G'' - S$ has size at most d and belongs to $\mathcal{P}$.

The overall construction of G' and G'' will display m *almost* copies of the encoding of an *edgeless* G arranged in a cycle. Each copy embeds one distinct edge of G. The point of having the information of G distilled edge by edge in G' and G'' is to control the treewidth. This general idea originates from a paper of Lokshtanov et al. [12].

Construction. We first describe G'. As a slight abuse of notation, a gadget (and, more generally, a subpart of the construction) may refer to either a subset of vertices or to an induced subgraph. For each $e = \langle i^e, j^e\rangle\langle i'^e, j'^e\rangle \in E$, we detail the internal construction of H^e and S^e of Figure 1 and how they are linked to one another. Each vertex $v = \langle i, j\rangle$ of G is represented by a gadget $H^e(v)$ on $3d-2$ vertices in G': a path on $d-3$ vertices whose endpoints are v^e_{-a} and v^e_{-b}, an isolated vertex v^e_+, and two disjoint cycles of length d. Observe that if $d = 4$, then v^e_{-a} and v^e_{-b} is the same vertex. We add all the edges between $H^e(\langle i, j\rangle)$ and $H^e(\langle i, j'\rangle)$ for $i, j, j' \in [k]$ with $j \neq j'$. We also add all the edges between $H^e(\langle i^e, j^e\rangle)$ and $H^e(\langle i'^e, j'^e\rangle)$. We call H^e the graph induced by the union of every $H^e(v)$, for $v \in V(G)$. The *row/column selector* gadget S^e consists of a set S^e_r of k vertices with one vertex r^e_i for each row index $i \in [k]$, and a set S^e_c of k vertices with one vertex c^e_j for each column index $j \in [k]$. The gadget S^e forms an independent set of size $2k$. We arbitrarily number the edges of G: $e_1, e_2, \ldots, e_m$. For each $h \in [m]$ and $v = \langle i, j\rangle \in V$, we link $v^{e_h}_{-a}$ to $r^{e_h}_i$ (the row index of v) and $v^{e_h}_{-b}$ to $c^{e_h}_j$ (the column index of v). We also link, for every $h \in [m-1]$, $v^{e_h}_+$ to $r^{e_{h+1}}_i$ and to $c^{e_{h+1}}_j$, and $v^{e_m}_+$ to $r^{e_1}_i$ and to $c^{e_1}_j$. That concludes the construction (see Figure 2). To obtain G'' from G', we add the edges $c^{e_h}_j c^{e_h}_{j+1}$ for every $h \in [m]$ and $j \in [k-1]$. We ask for a deletion set S of size $s := (3d-2)k(k-1)m$.

Treewidth of G' and G''. For any edge $e \in E$, we set $H(e) := H^e(\langle i^e, j^e\rangle) \cup H^e(\langle i'^e, j'^e\rangle)$. For any $i \in [m-1]$, we set $\tilde{S}_i := S^{e_1} \cup S^{e_i} \cup S^{e_{i+1}}$, and $\tilde{S}_m := S^{e_1} \cup S^{e_m}$. For each $e \in E$, and $i \in [k]$, $H^e(i)$ denotes the union of the $H^e(v)$ for all vertices v of the i-th row. Here is a path decomposition of G' and G'':

$$\tilde{S}_1 \cup H(e_1) \cup H^{e_1}(1) \to \tilde{S}_1 \cup H(e_1) \cup H^{e_1}(2) \to \ldots \to \tilde{S}_1 \cup H(e_1) \cup H^{e_1}(k) \to$$
$$\vdots$$
$$\tilde{S}_m \cup H(e_m) \cup H^{e_m}(1) \to \tilde{S}_m \cup H(e_m) \cup H^{e_m}(2) \to \ldots \to \tilde{S}_m \cup H(e_m) \cup H^{e_m}(k).$$

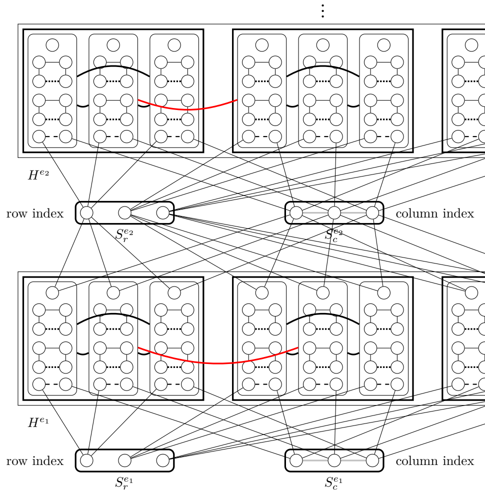

Figure 2 The overall picture of G' and G'' with $k = 3$. Dotted edges are subdivided $d-4$ times; if $d = 4$, they are simply edges. Dashed edges are subdivided $d-5$ times; if $d = 4$, the two endpoints are in fact a single vertex. Edges between two boxes link each vertex of one box to each vertex of the other box. The gray edges in the column selectors $S_c^{e_h}$ are only present in G''.

As, for any $h \in [m]$, $|\tilde{S}_h| \leqslant 6k$, $|H(e_h)| = 2(3d-2)$, and $|H^{e_h}(i)| \leqslant (3d-2)k$ for any $i \in [k]$, the size of a bag is bounded by $\max_{h\in[m],i\in[k]} |\tilde{S}_h \cup H(e_h) \cup H^{e_h}(i)| \leqslant 6k+2(3d-2)+(3d-2)k = (3d+4)k + 6d - 4$.

Correctness. If there is an independent set I of size k in G, a solution to a Bounded $\mathcal{P}$-Component Vertex Deletion or Bounded $\mathcal{P}$-Block Vertex Deletion instance can be obtained by deleting from each H^e every $H^e(v)$ such that $v \notin I$.

We show that $2 \Rightarrow 1$ and $3 \Rightarrow 1$. We assume that there is a set $S \subseteq V'$ of size at most s such that all the blocks of $G'' - S$ (resp. $G' - S$) have size at most d. We note that this corresponds to assuming condition 3 (resp. a weaker assumption than condition 2) holds. We show that there are at most $3d-2$ vertices of $H^e(i)$ remaining in $G'' - S$ (or $G' - S$). Assume, for the sake of contradiction, that $H^e(i) - S$ contains at least $3d-1$ vertices. Observe that $H^e(i) - S$ cannot contain at least one vertex from three distinct $H^e(u)$, $H^e(v)$, and $H^e(w)$ (with u, v and w in the i-th row of G), since then $H^e(i) - S$ would be 2-connected (and of size $> d$). For the same reason, $H^e(i) - S$ cannot contain at least two vertices in $H^e(u)$ and at least two vertices in another $H^e(v)$. Therefore, the only way of fitting $3d-1$ vertices in $H^e(i) - S$ is the $3d-2$ vertices of an $H^e(u)$ plus one vertex from some other $H^e(v)$. But then, this vertex of $H^e(v)$ would form, together with one C_d of $H^e(u)$, a 2-connected subgraph of $G'' - S$ (or $G' - S$) of size $d+1$. Now, we know that $|H^e(i) \cap S| \geqslant (3d-2)(k-1)$. As there are precisely mk sets $H^e(i)$ in G' (and they are disjoint), it further holds that $|H^e(i) \cap S| = (3d-2)(k-1)$, since otherwise S would contain strictly more than $s = (3d-2)k(k-1)m$ vertices. Thus, $H^e(i) - S$ contains exactly $3d-2$ vertices. By the previous remarks, $H^e(i) - S$ can only consist of the $3d-2$ vertices of the same $H^e(u)$ or $3d-3$ vertices of $H^e(u)$ plus one vertex from another $H^e(v)$. In fact, the latter case is not possible, since the vertex of $H^e(v)$ would form, with at least one remaining C_d of the $3d-3$ vertices of $H^e(u)$, a 2-connected subgraph of $G'' - S$ (or $G' - S$) of size $d+1$. This is why we needed two disjoint C_ds in the construction instead of just one. So far, we have proved that, assuming condition 2 or condition 3 holds, for any $e \in E$ and $i \in [k]$, $H^e(i) \cap S = H^e(v_{i,e})$ for some vertex $v_{i,e}$ of the i-th row of G, and for any $e \in E$, $S^e \cap S = \emptyset$.

In what follows, we show that $v_{i,e}$ does not depend on e. Formally, we want to show that there is a v_i such that, for any $e \in E$, $v_{i,e} = v_i$. Observe that it is enough to derive that, for any $h \in [m]$, $v_{i,e_h} = v_{i,e_{h+1}}$ (with $e_{m+1} = e_1$). Let $j \in [k]$ (resp. $j' \in [k]$) be the column of v_{i,e_h} (resp. $v_{i,e_{h+1}}$) in G. We first assume condition 2 holds. For any $h \in [m]$, ${v_{i,e_h}}^{e_h}_{+}$, $r_i^{e_{h+1}}$, $c_{j'}^{e_{h+1}}$, $c_j^{e_{h+1}}$ plus the path $P^{e_{h+1}}_{v_{i,e_{h+1}}}$ (between ${v_{i,e_{h+1}}}^{e_{h+1}}_{-a}$ and ${v_{i,e_{h+1}}}^{e_{h+1}}_{-b}$) induces a path (in particular, a connected subgraph) of size $d+1$ in $G'' - S$, unless $j = j'$ (with $e_{m+1} = e_1$). Therefore, $j = j'$. As v_{i,e_h} and $v_{i,e_{h+1}}$ have the same column j and the same row i in G, $v_{i,e_h} = v_{i,e_{h+1}}$. Showing the same property under 3 is done similarly. We can now safely define $v_i := v_{i,e}$ and conclude by proving that $\{v_1, v_2, \ldots, v_k\}$ is a clique. ◀

References

1 Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Optimal algorithms for hitting (topological) minors on graphs of bounded treewidth. *CoRR*, abs/1704.07284, 2017. `arXiv:1704.07284`.

2 Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inf. Comput.*, 243:86–111, 2015. `doi:10.1016/j.ic.2014.12.008`.

3 Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michal Pilipczuk. A c$^{\mathrm{k}}$ n 5-approximation algorithm for treewidth. *SIAM J. Comput.*, 45(2):317–378, 2016. `doi:10.1137/130947374`.

4 Édouard Bonnet, Nick Brettell, O-joung Kwon, and Dániel Marx. Parameterized vertex deletion problems for hereditary graph classes with a block property. In *Graph-Theoretic Concepts in Computer Science*, volume 9941 of *Lecture Notes in Comput. Sci.*, pages 233–244, 2016.

5 Édouard Bonnet, Nick Brettell, O-joung Kwon, and Dániel Marx. Generalized feedback vertex set problems on bounded-treewidth graphs: chordality is the key to single-exponential parameterized algorithms. *ArXiv e-prints*, 2017. arXiv:1704.06757.

6 Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990. `doi:10.1016/0890-5401(90)90043-H`.

7 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 150–159. IEEE Computer Society, 2011. `doi:10.1109/FOCS.2011.23`.

8 Reinhard Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, Heidelberg, fourth edition, 2010. `doi:10.1007/978-3-642-14279-6`.

9 Pål Grønås Drange, Markus S. Dregi, and Pim van 't Hof. On the computational complexity of vertex integrity and component order connectivity. *Algorithmica*, 76(4):1181–1202, 2016. `doi:10.1007/s00453-016-0127-x`.

10 Jessica Enright and Kitty Meeks. Deleting edges to restrict the size of an epidemic: A new application for treewidth. In Zaixin Lu, Donghyun Kim, Weili Wu, Wei Li, and Ding-Zhu Du, editors, *Combinatorial Optimization and Applications - 9th International Conference, COCOA 2015, Houston, TX, USA, December 18-20, 2015, Proceedings*, volume 9486 of *Lecture Notes in Computer Science*, pages 574–585. Springer, 2015. `doi:10.1007/978-3-319-26626-8_42`.

11 Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Efficient computation of representative sets with applications in parameterized and exact algorithms. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algo-*

rithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014, pages 142–151. SIAM, 2014. doi:10.1137/1.9781611973402.10.

12 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Known algorithms on graphs on bounded treewidth are probably optimal. In Dana Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 777–789. SIAM, 2011. doi:10.1137/1.9781611973082.61.

13 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Slightly superexponential parameterized problems. In Dana Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 760–776. SIAM, 2011. doi:10.1137/1.9781611973082.60.